

## REMARKS

Claims 1-18 and 20-29 are pending in the present application. Claim 19 has been canceled. Claims 1-8, 13, 14, 17, 18, 23-26, and 28 are amended. Reconsideration of the claims is respectfully requested.

### **I. 35 U.S.C. § 102, Anticipation**

The Office Action rejects claims 1-8, 13-18, 23-26, and 28 under 35 U.S.C. § 102 as being anticipated by *Gosling et al.* (EP 0 810 524 A1). This rejection is respectfully traversed.

With respect to claims 1, 13, and 17, the Office Action states:

As per claims 1, 13 and 17, Gosling teaches a method and computer program product for extending the capabilities of a web server, comprising the steps of:

- sending a request from a client to the web server, the request including an address for the code module needed to service the request (see pg. 2, lines 10-12, pg. 3, lines 39-48, and pg. 4, lines 5-15, Gosling teaches a thread that retrieves a client request in which the thread maps the request to a servlet name);
- if the code module is unavailable at the web server, having the web server use the address to request the code module from the publishing server (see pg. 3, lines 25-30);
- installing the code module at the server; and performing the request at the web server using the install code module (see pg. 3, lines 25-34, Gosling teaches a servlet being uploaded by the local server from a remote server, executing the servlet processing the client's request at the local server);
- serving a response to the request back to the client (see pg. 4, lines 34-40).

Office Action, dated February 14, 2003. *Gosling* teaches an apparatus and method for processing servlets in which a specified servlet object corresponding to a request may be uploaded from a remote server to the server receiving the request. The specified servlet object is then executed to obtain dynamically generated information corresponding to the request. See *Gosling*, page 2, lines 29-34. *Gosling* states:

A server administrator may specify that part of the client request is the name of the servlet, as found in an administered servlets directory. At many sites, that directory would be shared between servers which share the load of processing for the site's clients. Some servers may

be able to automatically invoke servlets to filter the output of other servlets, based on their administrative configuration. For example, particular types of servlet output may trigger post-processing by other servlets, perhaps to perform format conversions. Properly authorized clients may specify the servlet to be invoked, without administrative intervention.

*Gosling*, page 4, lines 10-15. Thus, in *Gosling*, a client request may specify a servlet to provide output for a request. In *Gosling*, the client may also somehow specify a servlet to be invoked to perform format conversions.

Representative claim 1, as amended, recites:

1. A method for extending the capabilities of a web server, comprising the steps of:
  - sending a request from a client to the web server, the request identifying requested content and including addresses for a plurality of code modules needed to service the request;
  - if a given code module is unavailable at the web server, having the web server use a corresponding address to request the given code module from a publishing server;
  - installing the given code module at the web server;
  - receiving the requested content at the web server; and
  - applying the plurality of code modules sequentially to the requested content.

The present invention allows the client to specify a plurality of code modules or servlets to be applied to the requested content. At best, *Gosling* teaches that a request may specify a first servlet to provide dynamically created output and *Gosling* also suggests that a client may somehow specify a second servlet that may be invoked to perform conversion on the output of the first servlet. However, *Gosling* does not teach or suggest a client sending a request that identifies requested content and includes addresses for a **plurality** of code modules needed to service the request and the server **applying the plurality of code modules sequentially** to the requested content, as recited in representative claim 1, as amended.

The applied prior art fails to teach or suggest each and every claim limitation; therefore, *Gosling* does not anticipate claim 1. Claims 13, 17, 23, and 28 recite subject matter addressed above with respect to claim 1 and are allowable for the same reasons. Since claims 2-9, 14-16, 18, and 24-26 depend from claims 1, 13, 17, and 23, the same

distinctions between *Gosling* and the invention recited in claims 1, 13, 17, and 23 apply for these claims. Consequently, it is respectfully urged that the rejection of claims 1-8, 13-18, 23-26, and 28 is overcome.

Furthermore, *Gosling* does not teach, suggest, or give any incentive to make the needed changes to reach the presently claimed invention. Absent the examiner pointing out some teaching or incentive to implement *Gosling* with a client request that identifies requested content and includes addresses for a plurality of code modules needed to service the request and the server applying the plurality of code modules sequentially to the requested content, one of ordinary skill in the art would not be led to modify *Gosling* to reach the present invention when the reference is examined as a whole. Absent some teaching, suggestion, or incentive to modify *Gosling* in this manner, the presently claimed invention can be reached only through an improper use of hindsight using the Applicant's disclosure as a template to make the necessary changes to reach the claimed invention.

## **II. 35 U.S.C. § 103, Obviousness**

The Office Action rejects claims 9, 12, 20-22, and 27 under 35 U.S.C. § 103 as being unpatentable over *Gosling* in view of *Murray* (US Patent No. 5,944,781). This rejection is respectfully traversed.

With respect to claim 9, the Office Action states:

As per claim 9, *Gosling* discloses the invention substantially as claimed, including:

- receiving a request from a client, the request identifying a code module required to process the request (see pg. 2, lines 10-12, pg. 3, lines 39-48, and pg. 4, lines 5-15, *Gosling* teaches a thread that retrieves a client request in which the thread maps the request to a servlet name); and
- responsive to a determination that the code module is not available at the web server (see pg. 3, lines 25-30).

However, *Gosling* does not explicitly disclose:

- uploading a code module from the client to the web server; and
- at the web server, using the uploaded code module to as needed to service a given request from the web client.

*Murray* discloses:

- uploading a code module from the client to the web server (col. 2, lines 54-65); and

- at the web server, using the uploaded code module to as needed to service a given request from the web client (col. 4, lines 39-45).

Therefore, one of ordinary skill in the art at the time the invention was made would have found it obvious to combine the teachings of Gosling and Murray to upload a code module from the client to the web server and at the web server using the code module as needed to service a given request from the client in Gosling allowing easy access to a code module servicing a request from the client.

Office Action, dated February 14, 2003. Applicant respectfully disagrees. As admitted in the Office Action, *Gosling* does not teach or suggest uploading a code module from a client to the server and, at the web server, using the uploaded code module to service the request from the client.

*Murray* teaches a persistent executable object system and method in which a client may post an applet and data to a server for persistent storage. Subsequently, the client may download the applet and data for execution. The cited portion of *Murray* states:

The client sends an HTTP POST message, the message typically used in the prior art for transmission of user form entries from the client to a server. The POST message begins with a URL that initiates an HTTP connection. The URL includes the address of the server, information identifying a remote database, the name of an object, and a particular storage or retrieval command for that object. If an object is to be transferred to the server, it follows the URL in the form of a serial data stream. If the object is to be transferred from the server to the client, the server responds to the POST message by transferring the object in the form of a serial data stream.

*Murray*, col. 2, lines 54-65. Thus, *Murray* teaches that the client may post an object to the server. However, *Murray* does not teach or suggest “receiving a request from a client, the request identifying a code module required to process the request,” as recited in claim 9. *Murray* also states:

The persistence capability is provided by adding an object storage capability at the server and a protocol for saving and retrieving objects at the server under the direction of the client. The applet viewer is modified at the client to operate in accordance with this saving and retrieving protocol. The modified applet viewer transmits

and receives objects by embedding them in serial form within HTTP messages.

*Murray*, col. 4, lines 39-45. Again, *Murray* teaches that a client may store and retrieve objects or applets at a server. The server in *Murray* is merely a storage receptacle for saving and retrieving objects. The server in *Murray* does not actually execute the applet in any way. Therefore, *Murray* does not teach or suggest, “at the web server, using the uploaded code module as needed to service a given request from the web client,” as recited in claim 9.

*Murray* does not make up for the deficiencies of *Gosling*. Even if a person of ordinary skill in the art would have been motivated to combine *Gosling* and *Murray*, the combination would not result in the present invention. The applied references, taken alone or in combination fail to teach or suggest each and every claim limitation. Therefore, the combination of *Gosling* and *Murray* does not render claim 9 obvious.

Claim 20 recites subject matter addressed above with respect to claim 9 and is allowable for the same reasons. Since claims 12, 21, and 22 depend from claims 9 and 20, the same distinctions between *Gosling* and *Murray* and the invention recited in claims 9 and 20 apply for these claims. Consequently, it is respectfully urged that the rejection of claims 9, 12, and 20-22 is overcome.

With respect to claim 27, the Office Action states:

As per claim 27, *Gosling* discloses the claimed invention substantially as claimed. However, *Gosling* does not explicitly disclose:

- means for deleting a code module from the server upon a given occurrence.

*Murray* discloses:

means for deleting a code module from the server upon a given occurrence (col. 5, lines 32-35).

Therefore, one of ordinary skill in the art at the time the invention was made would have found it obvious to combine the teachings of *Gosling* in view of *Murray* and *Fields* to delete a code module from the server upon a given occurrence in *Gosling* allowing the server to save memory by removing any code modules that are used only once.

Office Action, dated February 14, 2003. Applicant respectfully disagrees. *Murray* teaches that persistence capability allows a client to remove an applet from storage. However, *Murray* does not teach means for deleting a code module from the server,

wherein a code module is applied to requested content to perform a client request, as recited in claim 27. Furthermore, *Murray* also fails to teach or suggest a client sending a request that identifies requested content and includes addresses for a plurality of code modules needed to service the request and the server applying the plurality of code modules sequentially to the requested content. Therefore, *Murray* does not make up for the deficiencies of *Gosling*. The applied references, taken alone or in combination, fail to teach or suggest each and every claim limitation. Therefore, claim 27 is not rendered obvious by the combination of *Gosling* and *Murray*.

Therefore, the rejection of claims 9, 12, 20-22, and 27 under 35 U.S.C. § 103 is overcome.

The Office Action rejects claims 10 and 11 under 35 U.S.C. § 103 as being unpatentable over *Gosling* in view of *Murray* and further in view of *Fields et al.* (US Patent No. 6,412,008). This rejection is respectfully traversed.

With respect to claims 10 and 11, the Office Action states:

As per claim 10, *Gosling* in view of *Murray* discloses the claimed invention substantially as claimed.

However, *Gosling* in view of *Murray* does not explicitly disclose:

- wherein the web client is a pervasive computing client.

*Fields* discloses:

- wherein the web client is a pervasive computing client (col. 1, lines 48-51, and col.4, lines 15-18).

Therefore, one of ordinary skill in the art at the time the invention was made would have found it obvious to combine the teachings of *Gosling* in view of *Murray* and *Fields* wherein the web client is a pervasive client in *Gosling* in view of *Murray* because it would provide adequate means for efficiently sending and receiving information while in any given location.

As per claim 11, *Gosling* discloses:

- wherein the code module translates data into a given proprietary format and serves the translated data back to the pervasive computing client (pg. 4, lines 13-15, and lines 34-40, *Gosling* discloses that a servlet can perform format conversions and send the information back to the client).

Office Action, dated February 14, 2003. Applicant respectfully disagrees. *Fields* does not make up for the deficiencies of *Gosling* and *Murray*, as addressed above. As such, the applied references, taken alone or in combination, fail to teach or suggest each and

every claim limitation and the combination of *Gosling*, *Murray*, and *Fields* does not render claims 10 and 11 obvious.

Therefore, the rejection of claims 10 and 11 under 35 U.S.C. § 103 is overcome.

The Office Action rejects claim 29 under 35 U.S.C. § 103 as being unpatentable over *Gosling* in view of *Fields*. This rejection is respectfully traversed.

With respect to claim 29, the Office Action states:

As per claim 29, *Gosling* discloses the claimed invention substantially as claimed. However, *Gosling* does not explicitly disclose:

- wherein the web client is a pervasive computing client.

*Fields* discloses:

- wherein the web client is a pervasive computing client (col. 1, lines 48-51, and col. 4, lines 15-18).

Therefore, one of ordinary skill in the art at the time the invention was made would have found it obvious to combine the teachings of *Gosling* and *Fields* wherein the web client is a pervasive client in *Gosling* because it would provide adequate means for efficiently sending and receiving information while in any given location.

Office Action, dated February 14, 2003. Applicant respectfully disagrees. *Fields* does not make up for the deficiencies of *Gosling*, as addressed above. As such, the applied references, taken alone or in combination, fail to teach or suggest each and every claim limitation and the combination of *Gosling* and *Fields* does not render claim 29 obvious.

Therefore, the rejection of claim 29 under 35 U.S.C. § 103 is overcome.

**III. Conclusion**

It is respectfully urged that the subject application is patentable over the prior art of record and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: June 16, 2003

Respectfully submitted,

A handwritten signature in black ink, appearing to read 'S. Tkacs', is written over a horizontal line.

Stephen R. Tkacs  
Reg. No. 46,430  
Carstens, Yee & Cahoon, LLP  
P.O. Box 802334  
Dallas, TX 75380  
(972) 367-2001  
Agent for Applicants